**2017级机器学习方向第一次竞赛说明文档**

报告人：安博洋

**第一次尝试：**

在老师发布第一次竞赛，脉冲星候选物样本数据集的二分类问题时，当时，我们正好学完逻辑回归模型，这个模型恰好是用来解决二分类的问题。所以我第一想到的算法模型便是逻辑回归。逻辑回归算法的主要思想是用sigmod函数将二分类问题转换为概率大小的问题，具体做法是初始时给出一组参数，然后利用梯度下降的方法来训练出使得代价函数取最小的一组参数，参数训练出来后，将数据集带入，算出概率，当概率大于某个0.5时预测为一类，概率小于0.5时预测为另一类。于是，便开始利用代码实现，但当我实现完后，结果并没有我想象中的那么理想，正确率只有0.49047，我把预测结果分析了一下，发现预测的类别全是0，也就是算出的概率全部是小于0.5，于是我开始调整各个参数，把分类的概率调到0.005，调整学习率的大小，经过一系列的调整后，终于分类出的结果不再全是一个类别，但争取率既不稳定，也不高，最高只有0.789，最低是0.7023，于是，我便把这个模型暂时搁下，着手实验别的模型算法。

**第二次尝试：**

在我还不知道先从哪个模型入手时，我的一个同学用KNN算法（K近邻）预测出了正确率，高达0.8120，而且比较稳定。于是，我便抱着侥幸的心理准备也用代码实现一下KNN算法。KNN算法的主要思想是，先找到一个最合适的K值，然后选出距离每一个样本最近的K个样本，然后采用多数表决的方法，这K个样本中哪一类的数量最多，就把该样本也预测为那一类。具体流程是，首先需要规范化预处理训练样本的输入部分，把数据值都集中到某个数附近，然后基于欧式距离，并采用m折交叉验证的方式选择最优的K值（取最小平均错误率所对应的K值为最终的K值），最后利用测试集进行预测，得出正确率。但当我用代码实现后，算出的正确率并没有那么高，和之前的逻辑回归的争取率差不多，唯一好点的是它比较稳定，正确率稳定在0.75左右，于是，我便开始调整参数，调整K的范围，但经过很多次的调整，正确率依然没有超过0.8，于是，我又把KNN模型算法搁置，又回到最初的起点。

**第三次尝试：（一个奇妙的启发）**

在我的成果又回到起点后，正当我准备继续挑选另一个模型算法时，我忽然有了一个想法，第一次用逻辑回归模型时，由于这次竞赛用的数据集普遍比较大，导致最后用的参数使得最优的概率与0.5差距很大，我采取的措施是调小各种参数，比如分类概率值，学习率什么的，为的就是把最终算出的概率能够散布在0.5附近。但受到KNN算法的启发，它在工作前，先把数据集进行了规范化预处理，都整合到一个数的附近，使数据值变得很小，也很集中。那我同样也可以不去调整那些参数，而是把数据集进行规范化预处理，同样可以起到相同的效果。于是，我便重现改正逻辑回归算法，在学习训练之前，先把数据集进行了规范化预处理。果然，一下算出的概率同集中分布在0.5左右，可以起到很好的分类效果，正确率也一下上升到0.823，而且很稳定。于是，我便拿来老师的测试集数据集进行预测，并提交到Kaggle上，正确率达到了0.84285。

代码中各个函数的功能简单介绍如下：（核心代码解释在代码.py 文件中以注释方式写出）

init\_data(data)

规范化预处理数据集

sigmod(z)

sigmod函数，将分类问题转换为概率问题

grad\_descent(data, target)

逻辑回归实现函数，并返回参数矩阵

judge(data, weights, alpha)

预测函数，对测试集进行类别的预测，alpha 是判断阈值

write\_data(filename\_test,forecast)

将预测出来的类别写到一个文件中